6566/6567 VIDEO INTERFACE CHIPS

The 6566/6567 are multi purpose video controller devices for use in both computer video terminals and video game applications. The various operating modes and options within each mode are described.

CHARACTER DISPLAY MODE

In the character display mode, the 6566/6567 fetches CHARACTER POINTERs from the VIDEO MATRIX area of memory and translates the pointers to character dot location addresses in the 2,048 byte CHARACTER BASE area of memory. The video matrix is comprised of 1,000 consecutive locations in memory which each contain an eight bit character pointer. The location of the video matrix within memory is defined by VM13-VM10 in register 24 ($18) which are used as the 4 MSB of the video matrix address. The lower order 10 bits are provided by an internal counter which steps through the 1000 character locations. Note that the 6566/6567 provides 14 address outputs; therefore additional system hardware may be required for complete system emeory decodes. In addition to the 8 bit character pointer, a 4-bit COLOR NYBBLE is associated with each video matrix location (the video matrix memory must be 12 bits wide). The eight bit character pointer permits up to 256 different character definations to be available simultaneously. Each character is an 8x8 dot matrix stored in the character base as eight consecutive bytes of 8 bits. The location of the character base is defined by CB13-CB11 also in register 24 ($18) which are used for the 3 most significant bits (MSB) of the character base address. The 11 lower order addresses are formed by the 8 bit character pointer from the video matrix (CB10-CB3) which points to a particluar character and a 3 bit raster counter (CB2-CB0) which selects one of the eight character bytes. The resulting characters are formated as 25 rows of 40 characters each.

STANDARD CHARACTER MODE

In the standard character mode, the 8 sequential bytes from the character base are displayed directly on the 8 lines in each character region. A "0" bit causes the background #0 color (from register 33 ($21))to be displayed while the color selected by the color nybble is displayed for a "1" bit (see the Color Code Table). Therefore, each character has a unique color determined by the 4-bit color nybble (1 of 16) and all characters share the common background color.

MULTICOLOR CHARACTER MODE

The multi color mode is selected by setting the MCM bit in register 22 ($16) to "1". This causes the dot data stored in the character base to be interpreted in a different manner. If the MSB of the color nybble is a "0", the character will be displayed as described in standard character mode (however, only the lower order 8 colors are available). When the MSB of the color nybble is a "1" the character bits are interpreted as follows:

A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
CB13	VM9	VM8	VM7	VM6	VM5	VM4	VM3	VM2	VM1	VM0	RC2	RC1	RC0

VMx denotes the video matrix counter outputs, RCx denotes the 0-7 raster counter and CB13 is from register 24 ($18). The raster counter increments once for each horizontal raster line, while the video matrix counter increments once every eight rasters. This addressing results in each eight sequential memory locations being displayed as an 8x8 dot block on the video output.

STANDARD BIT MAP MODE

When standard bit map mode is in use, the color information is derived only from the data stored in the video matrix (the color nybble is disregarded). The 8 bits are divided into two 4-bit nybbles which determine the two colors available in the 8x8 dot block. When a bit in the display memory is a "0" the color of the output dot is set by the least significant nybble (LSN). Similarly, a display memory bit of "1" selects the output color determined by the MSN.

MULTI-COLOR BIT MAP MODE

Multi-colored bit map mode is selected by setting the MCM bit in register 22 ($16) to a "1" in conjunction with the BMM bit. Multi-color mode uses the same memory access sequence as standard bit map mode, but interprets the dot data as follows:

BIT PAIR	DISPLAY COLOR
00	Background #0 color (register 33 ($21))
01	Upper nybble of video matrix pointer
10	Lower nybble of video matrix pointer
11	Video matrix color nybble

Note that the color nybble (DB11-DB8) is used for the multi-color bit map mode. Again, since two bits are used to select one color, the horizontal dot size is doubled. Utilizing multi-color bit map mode up to 4 colors can be displayed in each 8x8 block.

MOVABLE OBJECT BLOCKS

The movable object block (MOB) is a special type of character which can be displayed at any one position on the screen without the block constraints inherent in character and bit map mode. Up to 8 unique MOBs can be displayed simultaneously, each defined by 63 bytes in memory which are displayed as a 24x21 dot array (shown below). A number of special features make MOBs especially suited for video graphics and game applicatons.

BIT PAIR	COLOR DISPLAYED
00	Background #0 Color (register 33 ($21))
01	Background #1 Color (register 34 ($22))
10	Background #2 Color (register 35 ($23))
11	Color specified by 3 LSB of color nybble

Since two bits are required to sepcify one color, the character is now displayed as a 4 x 8 matrix with each "dot" twice the horizontal size as in standard mode. Note, however, that each character can now contain 4 different colors.

EXTENDED COLOR MODE

The extended color mode allows the selection of individual background colors for each character region. This mode is selected by setting the ECM bit of register 17 ($11) to "1". The character dot data is displayed as in the standard mode, but the 2 MSB of the character pointer are used to select the background color for each character region as follows:

BIT PAIR	BACKGROUND COLOR DISPLAYED
00	Background #0 color (register 33 ($21))
01	Background #1 color (register 34 ($22))
10	Background #2 color (register 35 ($23))
11	Background #3 color (register 36 ($24))

Since the two MSB of the character pointers are used for color information, only 64 different character definations are available. The 6566/6567 will force CB10 and CB9 to "0" regardless of the original pointer values, so that only the first 64 character definations will be accessed.

NOTE - Extended color mode and multi-color mode can not be enabled simultaneously.

BIT MAP MODE

In bit map mode, the 6566/6567 fetches data from memory in a slightly different fashion, so that a one-to-one correspondence exists between each displayed dot and a memory bit. The bit map mode provides a screen resolution of 320x200 (XxY) individually controled display dots. Bit map mode is selected by setting the BMM bit in register 17 ($11) to a "1". The VIDEO MATRIX is still accessed as in character mode, but the video matrix data is no longer interpreted as addresses for characters, but rather as color data. Instead, the VIDEO MATRIX COUNTER is also used as an address to fetch the dot data for display from 8,000 byte DISPLAY BASE. The display base address is formed as follows:

MOB DISPLAY BLOCK

BYTE	BYTE	BYTE
00	01	02
03	04	05
.	.	.
.	.	.
.	.	.
57	58	59
60	61	62

ENABLE

Each MOB can be selectively enabled by setting its corresponding enable bit (MnE) to "1" in register 21 ($15).

POSITION

Each MOB is positioned via its X and Y position register (see register map) with a resolution of 512 horizontal and 256 vertical positions. The position of a MOB is determined by the upper-left corner of the array. X locations 23 to 347 ($17-$157) and Y locations 50 to 249 ($32-$F9) are visible. Therefore, not all available MOB positions are entirely visible on the screen.

COLOR

Each MOB has a seperate 4-bit register to determine the MOB color. The two MOB color modes are:

STANDARD MOB

In the standard mode, a "0" bit of MOB data is displayed as background #0 color and a "1" bit is displayed as the MOB color determined by the corresponding MOB Color register.

MULTI-COLOR MOB

Each MOB can be individually selected as a multi-color MOB via MnMC bits in the MOB Multi-color register 28 ($1C). When the MnMC bit is "1", the corresponding MOB is displayed in the multi-color mode. In the multi color mode, the MOB data is interpreted in pairs (similar to the other multi-color modes) as follows:

BIT PAIR	COLOR DISPLAYED
00	Backgrond #0 color (register 33 ($21))
01	MOB Multi-color #0 (register 37 ($25))
10	MOB Multi-color #1 (register 38 ($26))
11	MOB Color (registers 39-46 ($27-$2E))

Since two bits of data are required for each color, the resolution of the MOB is reduced to 12x21, with each horizontal "dot" expanded to twice standard size so that the overall MOB size does not change. Note that up to 3 colors can be displayed in each MOB (in addition to background #0) but that two of the colors are shared among all 8 MOBs in the multi-color mode.

MAGNIFICATION

Each MOB can be selectively expanded (2X) in both the horizontal and vertical directions. Two registers contain the control bits (MnXE,MnYE) for the magnification control:

RESIGTER	FUNCTION
29 ($1D)	Horizontal expand MnXE = "1"-expand; "0"-normal
23 ($17)	Vertical expand MnYE = "1"-expand; "0"-normal

When MOBs are expanded, no increase in resolution is realized. The same 24x21 array (12x21 if multi-colored) is displayed, but the size if each "dot" is doubled in the desired direction (up to 4X standard dot dimension if a MOB is both multi-colored and expanded).

PRIORITY

The priority of each MOB may be individually controled with respect to the other displayed information (from character or bit map modes). The priority of each MOB is set by the corresponding bit (MnDP) of register 27 ($1B) as follows:

BIT	PRIORITY
0	MOB will be displayed instead of any data (in front).
1	MOB Will be displayed only instead of Bkgd #0 or muli-color bit pair 01 (behind).

MOB data bits of "0" will always permit any other information to be displayed (i.e. MOBs may have "transparent sections).

The MOBs have a fixed priority with respect to each other, with MOB 0 having the highest priority and MOB 7 the lowest. When MOB data (except background #0) of two MOBs are co-incident, the data from the lower number MOB will be displayed.

COLLISION DETECTION

Two types of MOB collision (co-incidence) are detected, MOB to MOB collision and MOB to display data collision.

A collision between two MOBs occurs when non-background output data of two MOBs are co-incident. Co-incidence of MOB background (transparent) areas will not generate a collision. When a collision occurs, the MOB bits (MnM) in the MOB-MOB COLLISION register 30 ($1E) will be set to "1" for both colliding MOBs. As a collision between two (or more) MOBs occurs, the MOB-MOB collision bit for each collided MOB will be set. The collision bits remain set until a read of the collision register,

when they are automatically cleared. MOBs collisions are detected even if not visible.

The second type of collision is a MOB-DATA collision. A second register, MOB-DATA COLLISION register 31 ($1F) also has a bit (MnD) for each MOB which is set to "1" when both the MOB and display data are co-incident. Again, the co-incidence of only background data does not generate a collision. For special applications, the display data from the 0-1 multicolor bit pair also does not cause a collision. This feature permits their use as background display without interfering with true MOB collisions. A MOB-DATA collision can occur off-screen in the horizontal direction if actual display data has been scrolled to an off-screen position (see scrolling). The MOB-DATA COLLISION register also automatically clears when read.

The collision interrupt latches are set whenever the first bit of either register is set to "1". Once any collision bit within a register is set high, subsequent collisions will not set the interrupt latches until that collision register has been cleared to all "0"s by a read.

MOB MEMORY ACCESS

The data for each MOB is stored in 63 consecutive bytes of memory. Each block of MOB data is defined by a MOB pointer, located at the end of the VIDEO MATRIX. Only 1,000 bytes of the video matrix are used in the normal display modes, allowing the video matrix locations 1016-1023 ($3F8-$3FF) to be used for MOB pointers 0-7, respectively. The eight bit MOB pointer from the video matrix together with the six bits from the MOB byte counter (to keep track of 63 bytes) define the entire 14-bit address field:

A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	MC5	Mc4	Mc3	MC2	MC1	MC0

Where MPx are the MOB pointer bits from the video matrix and MCx are the internally generated MOB counter bits. The MOB pointers are read from the video matrix at the end of every raster line. When the .Y position register of a MOB matches the current raster count, the actual fetches of MOB data begin. Internal counters automatically step through the 63 bytes of MOB data, displaying three bytes on each raster.

OTHER FEATURES

SCREEN BLANKING

The entire display screen may be blanked by setting the BLNK bit in register 17 (#11) to a "1". When the screen in blanked, the entire screen will be filled with the exterior color as set in register 32 (#20). When blanking is enabled, only transparent (Phase 1) memory accesses are required, permitting full processor utilization of the system bus.

ROW/COLUMN SELECT

The normal display consists of 25 rows of 40 characters (or character regions) per row. For special display purposes, the display window may be reduced to 24 rows and 39 characters. There is no change in the format of the displayed information, except that characters (bits) adjacent to the border area will now be covered by the border. The select bits operate as follows:

RSEL	Number of rows	CSEL	Number of columns
0	24 rows	0	39 columns
1	25 rows	1	40 columns

The RSEL bit is in register 17 (#11) and the CSEL bit is in register 22 (#16). For standard display the larger display window is normally used, while the smaller display window is normally used in conjunction with scrolling.

SCROLLING

The display data may be scrolled up to one entire character space in both the horizontal and vertical direction. When used in conjunction with the smaller display window (above), scrolling can be used to create a smooth motion of display data while updating the system memory only when a new character row (or column) is required. Scrolling is also used to center a fixed display within the display window.

BITS	REGISTER	FUNCTION
X2,X1,X0	22 (#16)	Horizontal Position
Y2,Y1,Y0	17 (#11)	Vertical Position

LIGHT PEN

The light pen input latches the current screen position into a pair of registers (LPX,LPY) on a low-going edge. The X position register 19 (#13) will contain the 8 MSB of the X position at the time of transition. Since the X position is defined by a 512 counter (9 bits) resolution to 2 horizontal dots is provided. Similarly, the Y position is latched in its register 20 (#14) but here 8 bits provide single raster resolution within the visible display. The light pen latch may be triggered only once per frame, and subsequent triggers

within the same frame will be ignored.

RASTER REGISTER

The raster register is a dual function register. A read of the raster register 18 (#12) returns the lower 8 bits of the current raster positon (the MSB-RC8 islocated in register 17 (#11)). A write to the raster bits (including RC8) is latched for use in an internal raster compare. When the current raster matches the written value, the raster interrupt latch is set. The raster register can be interrogated to implement display changes outside the visible area to prevend display flicker. The display window is from raster 51 through raster 251 (#33-#FB).

INTERRUPT REGISTER

The interrupt register shows the status of the four sources of interrupt. An interrupt latch in register 25 (#19) is set to "1" when an interrupt source has generated an interrupt request. There are four sources of interrupt:

LATCH BIT	ENALBE BIT	WHEN SET
IRST	ERST	Set when (raster count) = (latched raster count)
IMDC	EMDC	Set by MOB-DATA collision register (first collision only)
IMMC	EMMC	Set by MOB-MOB collision register (first collision only)
ILP	ELP	Set by negative transition of LP input (once per frame)
IRQ		Set high by latch set and enabled (invert of IRQ/ output)

To enable an interrupt request to set the IRQ/ output to "0", the corredponding interrupt enable bit in register 26 (#1A) must be set to "1". Once an interrupt latch has been set, the latch may be cleared only by writing a one to the desired latch in the interrupt register. This feature allows selective handling of video interrupts without software required to "remember" active interrupts.

DYNAMIC RAM REFRESH

A dynamic ram refresh controller is built in to the 6566/6567 devices. Five 8-bit row addresses are refreshed every raster line. This rate guarantees a maximum delay of 2.02 ms between the refreshes of any single row address in a 128 refresh scheme (3.66 ms for a 256 refresh addresses). This refresh is totally transparent to the system, since the refresh occurs during Phase 1 of the system clock.

RESET

The reset bit (RES) in register 22 (#16) is not used for normal operation. Therefore it should be set to "0" when initializing the video chip. When set to a "1", the entire operation of the video chip is suspended, including video out and sync, memory refresh, and system bus access.

THEORY OF OPERATION

SYSTEM INTERFACE

The 6566/6567 video controller devices interact with the system data bus in a special way. A 65XX system requires requires the system busses only during the Phase 2 (clock high) portion of the cycle. The 6566/6567 devices take advantage of this feature by normally accessing system memory during the Phase 1 (clock low) portion of the clock cycle. Therefore, operations such as character data fetches and memory refresh are totally transparent to the processor and do not reduce the processor through-put. The video chips provide the interface control signals required to maintain this bus sharing.

The video devices provide the signal AEC (address enable control) which is used to disable the processor address bus drivers. AEC is active low which permits direct connection to the AEC inputs of the 65XX family. The AEC signal is normally activated during Phase 1, so that processor operation is not affected. Because of this bus "sharing", all memory accesses must be completed in 1/2 cycle. Since the video chips provide a 1Mhz clock (which must be used as system Phase 0), a memory cycle is 500ns including address setup, data access and data setup to the reading device.

Certain operations of the 6566/6567 require data at a faster rate than available by reading only during the Phase 1 time: specifically, the access of character pointers from the video matrix and the fetch of MOB data. Therefore, the processor must be disabled and the data accessed during the Phase 2 clock. This is accomplished via the BA (bus available) signal. The BA line is normally high but is brought low during Phase 1 to indicate that the video chip will require a Phase 2 data access. Three Phase 2 times are allowed after BA low for the processor to complete any current memory accesses. On the fourth Phase 2 after BA low, the AEC signal will remain low during Phase 2 as the video chip fetches data. The BA line is normally connected to the RDY input of a 65XX processor. The character pointer fetches occur every eighth raster during the display window and require 40 consecutive Phase 2 accesses. The MOB data fetches require 3 fetches as follows:

PHASE	DATA	CONDITION
1	MOB Pointer	Every raster
2	MOB Byte 1	Each raster while MOB is active
1	MOB Byte 2	Each raster while MOB is active
2	MOB Bbyte3	Each raster while MOB is active

The MOB pointers are fetched every other Phase1 at the end of each raster line. As needed, additional cycles will be used for MOB data fetches. All necessary bus control is provided by the 6566/6567 devices.

PROCESSOR INTERFACE

Aside from the special memory accesses described above, the 6566/6567 registers can be accessed similar to another peripheral

device. The following processor interface signals are provided:

DATA BUS (D87-D80)

The eight data bus pins are the bi-directional data port, cntrolled by CS/, RW, and Phase Out. The data bus can only be accessed while AEC is high.

CHIP SELECT

The chip select pin, CS/, is brought low to enable access to the device registers in conjunction with the address and RW pins. CS/ low is only valid while AEC is high.

READ/WRITE

The read/write input, R/W, is used to determine the direction of data transfer on the data bus, in conjunction with CS/. When R/W is high ("1") data is transfered from the selected register to the data bus output. When R/W is low ("0") data presented on the data bus pins is loaded into the selected register.

ADDRESS BUS

The lower six address pins, A5-A0, are bi-directional. During a processor read or write of the video device, these address pins are inputs. The data on the address inputs selects the register for read or write as defined in the register map.

CLOCK OUT

The clock output, Phase Out, is the 1Mhz clock used as the 65XX processor Phase 0 in. All system bus activity is referenced to this clock. The clock frequency is determined by dividing the 8Mhz video input clock by eight.

INTERRPUTS

The interrupt output, IRQ/, is brought low when an enabled source of interrupt occurs with in the device. The IRQ/ output is open drain, requiring an external pull-up resistor.

VIDEO INTERFACE

The video output signal from the 6566/6567 consists of two signals which must be extranally mixed together. SYNC/LUM output contains all the video data, including horizontal and vertical syncs, as well as the luminance information of the video display. SYNC/LUM is open drain, requiring an external pullup of 500 ohms. The COLOR output contains all the chrominance information, including the color reference burst and the color of all display data. The COLOR output is open source and sould be terminated with 1,000 ohms to ground. After appropriate mixing of these two signals, the resulting signal can directly drive a video monitor or be fed to a modulator for use with a standard television.

AEC	PK0	CS/	R/W	ACTION
0	0	X	X	PHASE 1 FETCH, REFRES
0	1		X	PHASE 2 FETCH <PROSESSOR OFF>
1	0	X	X	NO ACTION
1	1	0	0	WRITE TO SELECTED REGISTER
1	1	0	1	READ FROM SELECTED REGISTER
1	1	1	X	NO ACTION

SUMMARY OF 6566/6567 BUS ACTIVITY

REGISTER MAP

ADDRESS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DESCRIPTION
0 ($00)	M0X7	M0X6	M0X5	M0X4	M0X3	M0X2	M0X1	M0X0	MOB 0 X-position
1 ($01)	M0Y7	M0Y6	M0Y5	M0Y4	M0Y3	M0Y2	M0Y1	M0Y0	MOB 0 Y-position
2 ($02)	M1X7	M1X6	M1X5	M1X4	M1X3	M1X2	M1X1	M1X0	MOB 1 X-position
3 ($03)	M1Y7	M1Y6	M1Y5	M1Y4	M1Y3	M1Y2	M1Y1	M1Y0	MOB 1 Y-position
4 ($04)	M2X7	M2X6	M2X5	M2X4	M2X3	M2X2	M2X1	M2X0	MOB 2 X-position
5 ($05)	M2Y7	M2Y6	M2Y5	M2Y4	M2Y3	M2Y2	M2Y1	M2Y0	MOB 2 Y-position
6 ($06)	M3X7	M3X6	M3X5	M3X4	M3X3	M3X2	M3X1	M3X0	MOB 3 X-position
7 ($07)	M3Y7	M3Y6	M3Y5	M3Y4	M3Y3	M3Y2	M3Y1	M3Y0	MOB 3 Y-position
8 ($08)	M4X7	M4X6	M4X5	M4X4	M4X3	M4X2	M4X1	M4X0	MOB 4 X-position
9 ($09)	M4Y7	M4Y6	M4Y5	M4Y4	M4Y3	M4Y2	M4Y1	M4Y0	MOB 4 Y-position
0 ($0A)	M5X7	M5X6	M5X5	M5X4	M5X3	M5X2	M5X1	M5X0	MOB 5 X-position
1 ($0B)	M5Y7	M5Y6	M5Y5	M5Y4	M5Y3	M5Y2	M5Y1	M5Y0	MOB 5 Y-position
2 ($0C)	M6X7	M6X6	M6X5	M6X4	M6X3	M6X2	M6X1	M6X0	MOB 6 X-position
3 ($0D)	M6Y7	M6Y6	M6Y5	M6Y4	M6Y3	M6Y2	M6Y1	M6Y0	MOB 6 Y-position
4 ($0E)	M7X7	M7X6	M7X5	M7X4	M7X3	M7X2	M7X1	M7X0	MOB 7 X-position
5 ($0F)	M7Y7	M7Y6	M7Y5	M7Y4	M7Y3	M7Y2	M7Y1	M7Y0	MOB 7 Y-position
6 ($10)	M7X8	M6X8	M5X8	M4X8	M3X8	M2X8	M1X8	M0X8	MSB of X-position
7 ($11)	RC8	ECM	BMM+	BLNK	RSEL	Y2	Y1	Y0	See text
8 ($12)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	Raster register
9 ($13)	LPX8	LPX7	LPX6	LPX5	LPX4	LPX3	LPX2	LPX1	Light Pen X
0 ($14)	LPY7	LPY6	LPY5	LPY4	LPY3	LPY2	LPY1	LPY0	Light Pen Y
1 ($15)	M7E	M6E	M5E	M4E	M3E	M2E	M1E	M0E	MOB Enable
2 ($16)	-	-	RES	MCM	CSEL	X2	X1	X0	See text
3 ($17)	M7YE	M6YE	M5YE	M4YE	M3YE	M2YE	M1YE	M0YE	MOB Y-XEpand
4 ($18)	VM13	VM12	VM11	VM10	CB13	CB12	CB11	-	Memory Pointers
5 ($19)	IRQ/	-	-	-	ILP	IMMC	IMBC	IRST	Interrupt Register
6 ($1A)	-	-	-	-	ELP	EMMC	EMBC	ERST	Enable Interrupt
7 ($1B)	M7DP	M6DP	M5DP	M4DP	M3DP	M2DP	M1DP	M0DP	MOB-DATA Priority
8 ($1C)	M7MC	M6MC	M5MC	M4MC	M3MC	M2MC	M1MC	M0MC	MOB Multicolor Sel
9 ($1D)	M7XE	M6XE	M5XE	M4XE	M3XE	M2XE	M1XE	M0XE	MOB X-expand
0 ($1E)	M7M	M6M	M5M	M4M	M3M	M2M	M1M	M0M	MOB-MOB Collision
1 ($1F)	M7D	M6D	M5D	M4D	M3D	M2D	M1D	M0D	MOB-DATA Collision
2 ($20)	-	-	-	-	EC3	EC2	EC1	EC0	Exterior Color
3 ($21)	-	-	-	-	B0C3	B0C2	B0C1	B0C0	Bkgd #0 Color
4 ($22)	-	-	-	-	B1C3	B1C2	B1C1	B1C0	Bkgd #1 Color
5 ($23)	-	-	-	-	B2C3	B2C2	B2C1	B2C0	Bkgd #2 Color
6 ($24)	-	-	-	-	B3C3	B3C2	B3C1	B3C0	Bkgd #3 Color
7 ($25)	-	-	-	-	MM03	MM02	MM01	MM00	MOB Multicolor #0
8 ($26)	-	-	-	-	MM13	MM12	MM11	MM10	MOB Multicolor #1
9 ($27)	-	-	-	-	M0C3	M0C2	M0C1	M0C0	MOB 0 Color
0 ($28)	-	-	-	-	M1C3	M1C2	M1C1	M1C0	MOB 1 Color
1 ($29)	-	-	-	-	M2C3	M2C2	M2C1	M2C0	MOB 2 Color
2 ($2A)	-	-	-	-	M3C3	M3C2	M3C1	M3C0	MOB 3 Color
3 ($2B)	-	-	-	-	M4C3	M4C2	M4C1	M4C0	MOB 4 Color
4 ($2C)	-	-	-	-	M5C3	M5C2	M5C1	M5C0	MOB 5 Color
5 ($2D)	-	-	-	-	M6C3	M6C2	M6C1	M6C0	MOB 6 Color
6 ($2E)	-	-	-	-	M7C3	M7C2	M7C1	M7C0	MOB 7 Color

COLOR CODES

D4	D3	D1	D0	HEX	COLOR
0	0	0	0	0	BLACK
0	0	0	1	1	WHITE
0	0	1	0	2	RED
0	0	1	1	3	CYAN
0	1	0	0	4	PURPLE
0	1	0	1	5	GREEN
0	1	1	0	6	BLUE
0	1	1	1	7	YELLOW
0	0	0	0	8	BLACK
1	0	0	1	9	BROWN
1	0	1	0	A	LT RED
1	0	1	1	B	DARK GREY
1	1	0	0	C	MED GREY
1	1	0	1	D	LT GREEN
1	1	1	0	E	LT BLUE
1	1	1	1	F	LT GREY

generously contributed by
michael tomczyk

scanned by: www.commodore.international